IE Interfaces
Vol. 12, No. 1, pp. 19-25, March 1999.

Time Slot Scheduling Algorithm for SS/TDMA Networks with Intersatellite Links†

Kun-Nyeong Chang
Department of Management Information System, Silla University

위성간 링크를 가진 SS/TDMA 망에 대한 타임슬롯 스케쥴링 알고리즘

장근녕

The time slot scheduling problem for a satellite cluster with an arbitrary number of satellite is considered, which is one of the most interesting problems in the satellite communication scheduling area. This problem is known to be NP-complete, and several heuristic algorithms have been proposed. In this paper, a new efficient algorithm is suggested, which has lower computational complexity and provides much better solution than other existing algorithms.

1. Introduction

A satellite switched time division multiple access (SS/TDMA) system consists of a control unit to supervise system operations, a multibeam antenna to cover several geographically distributed zones, and an on-board switch to provide connections between the uplink and downlink beams according to the TDMA frame. The TDMA frame is divided into several smaller intervals called time slots, and each time slot is transmitted according to a configuration of the on-board switch corresponding to it. The objective is to accomplish the transmission of a given traffic load with maximum transponder utilization. SS/TDMA systems with single satellite were extensively studied under various optimization criteria [2,3,5-9,11].

However, in many practical situations, an SS/TDMA system has more than one satellite and intersatellite links (ISL) connecting two satellites, creating a satellite communication network. Intersatellite link has the additional advantage of allowing several small and less expensive satellites to join their coverage and capability so to have the communication power of a much larger and more expensive satellite [1]. Indeed, the INTELSAT 6 series are the first commercial satellites to employ SS/TDMA technique, and NASA's advanced com- munication technology satellite (ACTS) will use this technique.

In this case, it is necessary to solve the time slot scheduling problem for a satellite cluster which has to consider the ISL capacity constraints in addition to the single satellite SS/TDMA scheduling constraints. This problem with an arbitrary number of satellites is known to be NP-complete, which is intrinsically intractable and can thus be optimally solved only by algorithms that run in a time which grows as an exponential function in the size of the traffic matrix [1,10].

Several heuristic algorithms have been presented to obtain a near optimal solution of the time slot

†이 연구는 1998년도 신라대학교 연구비로 이루어졌음.

scheduling problem. Bertossi *et al.* [1] presented heuristic algorithms for a system with two satellites, each covering similar number of disjoint ground stations, and one ISL. Ganz *et al.* [4] presented a heuristic algorithm for a quite generalized system. The algorithm is based on the algorithm for open shop scheduling problem, and has $O(rM^2)$ computational complexity, where r is the number of non-zero elements in the traffic matrix and M is the number of geographical zones. Recently, for a generalized model, Kim *et al.* [10] suggested a new efficient algorithm with $O(rM^2)$ computational complexity, where each satellite has an arbitrary number of transponders. However, the solution quality and computational complexity of these algorithms are not satisfactory.

In this paper, a simple algorithm to solve the time slot scheduling problem for a satellite cluster is suggested. The main idea in the algorithm is to sort ordered uplink and downlink zone pairs by two key factors, traffic density and transponder/ISL capacity, and to sequentially select zone pairs satisfying transponder/ISL capacity constraints. Simulation results show that the proposed algorithm generates much better solutions than other compared algorithms. Furthermore, the computational complexity of the proposed algorithm is $O(r^2)$, which is smaller than the previous algorithms.

This paper is organized as follows. The problem formulation and a theoretical lower bound on the switching duration are given in section 2. Section 3 presents a heuristic algorithm. In section 4, extensive computational test results of the proposed algorithm are reported and compared with other existing algorithms. Section 5 concludes the paper.

2. Problem Formulation

This section presents definitions and the formal formulation of the time slot scheduling problem for a satellite cluster, which are similar to [4] and [10].

A cluster consisting of S satellites, $C=\{1,2,\cdots,S\}$, and a set of M disjoint geographical zones, $Z=\{1,2,\cdots,M\}$ are considered. Satellite p in the cluster covers M_p geographical zones in Z_p which is a subset of Z. Each zone is assumed to be covered by only one satellite. Hence, $Z_p \cap Z_q = \emptyset$ for two different satellites p and q.

The transponder/ISL capacity is characterized by $S\times S$ matrix L with entry l_{pq} representing the number of intersatellite links from satellite p to satellite q if $p\neq q$, otherwise, the number of transponders in satellite p.

The traffic demand is characterized by an $M\times M$ matrix D with entry d_{ij} representing the amount of traffic from uplink beam (source zone) i to downlink beam (destination zone) j, measured in time slot units. And the following submatrix notations are used in the rest of the paper.

$D(p, p)$: the traffic between zones visible by satellite p ($M_p\times M_p$ submatrix of D)

$D(p, q)$: the traffic between zones visible by satellite p and zones visible by satellite q ($M_p\times M_q$ submatrix of D ($p\neq q$))

$D(p, \cdot)$: the traffic originating from zones in Z_p ($M_p\times M$ submatrix of D)

$D(\cdot, q)$: the traffic arriving to zones in Z_q ($M\times M_q$ submatrix of D)

Note that the transmission of the traffic in the intersatellite submatrix $D(p, q)$ requires both a transponder and an ISL simultaneously, and d_{ij} and d_{ji} should be equal to 0 if zones i and j are visible by satellite p and satellite q $(p\neq q)$, respectively, and the two satellites are not interconnected by ISL's.

The scheduling algorithm has to decompose the given traffic matrix D into distinct switching matrices, $D=\sum_{i=1}^{n} D_i$, where n denotes the number of switching configurations. Each matrix characterizes a particular switching configuration and its corresponding traffic load being switched without conflict. To obtain a conflict free scheduling, a switching matrix D_i must be an $M\times M$ matrix with at most one positive entry in each line, at most l_{pq} positive entries in each submatrix corresponding to $D(p, q)$ where $p\neq q$, and at most $l_{pp}(l_{qq})$ positive entries in each submatrix corresponding to $D(p, \cdot)$ $(D(\cdot, q))$, respectively. The largest entry in a switching matrix D_i dictates the switching duration of D_i denoted by $\overline{D_i}$. The total duration needed to schedule the complete traffic matrix D is given by $\overline{D}=\sum_{i=1}^{n} \overline{D_i}$. A schedule for D is optimal if it achieves maximum transponder utilization among all the possible schedules for D. It has been shown that this is equivalent to minimize the schedule length, $\overline{D}$ [2,5].

Bertossi *et al.* [1] considered a special case where there are two satellites, each has M_p transponders and one ISL. Ganz *et al.* [4] considered the case where there are an arbitrary number of satellites and each satellite p in the cluster has M_p transponders and an arbitrary number of ISL's. Kim *et al.* [10] considered the case where there are an arbitrary number of satellites and each satellite p has $l_{pp}(1 \le l_{pp} \le M_p)$ transponders and an arbitrary number of ISL's. This paper considers the same case as [10].

The following Theorem 1 gives the theoretical lower bound on the minimal duration. The sum of entries in the ith row of the traffic matrix D and the sum of entries in the jth column of D are denoted by r_i and c_j, respectively. Let $T(p,q)$ denote the amount of traffic in the submatrix $D(p,q)$, that is, the sum of all entries in $D(p,q)$, and let $T(p,\cdot)$ and $T(\cdot,q)$ denote the amount of traffic in the submatrix $D(p,\cdot)$ and $D(\cdot,q)$, respectively.

Theorem 1. [10] Any schedule for D has length not smaller than LB

$$\mathrm{LB} = \max \begin{bmatrix} \max_{1\le i\le M}\{r_i\} \\ \max_{1\le j\le M}\{c_j\} \\ \max_{1\le p,q\le S, p\ne q}\{\lceil T(p,q)/l_{pq}\rceil\} \\ \max_{1\le p\le S}\{\lceil T(p,\cdot)/l_{pp}\rceil\} \\ \max_{1\le q\le S}\{\lceil T(\cdot,q)/l_{qq}\rceil\} \end{bmatrix}$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x. ■

3. Time Slot Scheduling Algorithm

This section presents a simple algorithm for the optimal time slot scheduling problem known to be NP-complete [1]. The main idea in the algorithm is to sort ordered uplink and downlink zone pairs (i,j)'s by two key factors, traffic density and transponder/ISL capacity, influencing the efficiency of time slot scheduling, and to sequentially select pairs satisfying transponder/ISL capacity constraints.

The discussion of the proposed algorithm is started by defining a temporary matrix $V=(v_1, v_2, \cdots)^T$ where $v_k=(v_k^u, v_k^d, v_k^t)$. Here v_k^u and v_k^d represent uplink and downlink zones of the kth zone pair, respectively, and v_k^t represents the amount of traffic between the uplink zone v_k^u and the downlink zone v_k^d. This matrix is simply generated by transforming the traffic matrix D.

In order to obtain a good solution for the time slot scheduling problem, the proposed algorithm tries to preferentially assign time slots to zone pairs which have more traffic amount and less transponder/ISL capacity. To take into account this requirement, a function $C(v_k)$, representing the degree of difficulty of assigning time slots to zone pair (v_k^u, v_k^d), is defined by:

$$C(v_k) = w\frac{v_k^t}{d^*} + (1-w)\left(1 - \frac{l_{s(v_k^u)s(v_k^d)}}{l^*}\right)$$

where d^* and l^* are the largest entries in the traffic matrix D and the transponder/ISL capacity matrix L, respectively, and w is a weighting factor $(0 \le w \le 1)$, and $s(z)$ represents the satellite which covers the zone z.

The proposed algorithm is described below in detail.

<u>Algorithm MDFS (Maximum Degree First Scheduling)</u>

Step 0. (Initialization) Let D and L be a given $M\times M$ traffic matrix and a given $S\times S$ transponder/ISL capacity matrix, respectively. Let $\overline{k}$ be the number of non-zero elements of D. Using the traffic matrix D, make an ordered matrix $V=(v_1, v_2, \cdots, v_{\overline{k}})^T$ such that $v_k > v_{k+1}$ and $v_k^t > 0$. Set $i=1$.

Step 1. (Obtaining a switching matrix D_i)

(1.0) Set $t_p^u = l_{pp}$ and $t_p^d = l_{pp}$ for all $p \in C$ where t_p^u and t_p^d are the numbers of available transponders in satellite p for the uplink and downlink communications, respectively. Set $z_j^u=0$ and $z_j^d=0$ for all zone $j\in Z$, and set $e_k=0$ for $1\le k\le \overline{k}$. Set $k=1$, $n=0$ and $\overline{d}=\infty$.

(1.1) If $(z_{v_k^u}^u=0, z_{v_k^d}^d=0)$, $(t_{s(v_k^u)}^u>0, t_{s(v_k^d)}^d>0)$ and $(l_{s(v_k^u)s(v_k^d)}>0)$, then go to Step (1.1.0) Otherwise, go to Step (1.2).

(1.1.0) Set $n=n+1$, $x_n=v_k$, $e_k=1$, $z^u_{v^u_k}=1$ and $z^d_{v^d_k}=1$. If $v^t_k<\bar{d}$ then $\bar{d}=v^t_k$.

(1.1.1) Set $t^u_{s(v^u_k)}=t^u_{s(v^u_k)}-1$ and $t^d_{s(v^d_k)}=t^d_{s(v^d_k)}-1$, and set $l_{s(v^u_k)s(v^d_k)}=l_{s(v^u_k)s(v^d_k)}-1$ if $s(v^u_k)\neq s(v^d_k)$. Go to Step (1.2).

(1.2) If $k<\bar{k}$, then go to Step (1.1) with replacing $k+1$ to k. Otherwise, set $\bar{n}=n$ and form a switching matrix D_i using the matrix $X=(x_1,\cdots,x_{\bar{n}})^T$ and $\bar{d}$, and go to Step 2.

Step 2. (Making a new ordered matrix V)

(2.0) Set $x^t_n=x^t_n-\bar{d}$, for $1\leq n\leq \bar{n}$. Extract, from the matrix X, the element x_n such that $x^t_n=0$. Decrease $\bar{n}$ by the number of extrac- ted elements. Initialize the transponder/ISL capacity matrix $\{l_{ij}\}$. Set $k=1, n=1$ and $s=0$.

(2.1) If $e_k=1$, then go to Step (2.2) with $s+1$ replacing to s, otherwise, go to Step (2. 1.0).

(2.1.0) If $(n\leq\bar{n})$ and $(x_n>v_k)$, then go to Step (2.1. 1), otherwise, go to Step (2.1.2).

(2.1.1) Set $v_{k-s}=x_n, n=n+1$ and $s=s-1$ and go to Step (2.1.0).

(2.1.2) Set $v_{k-s}=v_k$ and go to Step (2.2).

(2.2) Set $k=k+1$. If $k\leq\bar{k}$, then go to Step (2.1). If $n\leq\bar{n}$, then set $v_{k-s}=x_n$, $n=n+1$ and $s=s-1$ until $n>\bar{n}$. Go to Step 3.

Step 3. (Termination) Set $\bar{k}=\bar{k}-s$. If $\bar{k}=0$ then terminate, otherwise, go to Step 1 with replacing $i+1$ to i. In the steps 0 and 2, $v_{k_1}>v_{k_2}$ (v_{k_1} precedes v_{k_2}) means one of the following cases:

a) $C(v_{k_1})>C(v_{k_2})$,

b) $C(v_{k_1})=C(v_{k_2})$ and $l_{s(v^{u_{k_1}})s(v^{d_{k_1}})}<l_{s(v^{u_{k_2}})s(v^{d_{k_2}})}$,

c) $C(v_{k_1})=C(v_{k_2})$, $l_{s(v^{u_{k_1}})s(v^{d_{k_1}})}=l_{s(v^{u_{k_2}})s(v^{d_{k_2}})}$ and $v^t_{k_1}>v^t_{k_2}$

d) $C(v_{k_1})=C(v_{k_2})$, $l_{s(v^{u_{k_1}})s(v^{d_{k_1}})}=l_{s(v^{u_{k_2}})s(v^{d_{k_2}})}$, $v^t_{k_1}=v^t_{k_2}$ and $d_{v^u_{k_1}v^d_{k_1}}>d_{v^u_{k_2}v^d_{k_2}}$,

e) $C(v_{k_1})=C(v_{k_2})$, $l_{s(v^{u_{k_1}})s(v^{d_{k_1}})}=l_{s(v^{u_{k_2}})s(v^{d_{k_2}})}$, $v^t_{k_1}=v^t_{k_2}$, $d_{v^u_{k_1}v^d_{k_1}}=d_{v^u_{k_2}v^d_{k_2}}$ and $(M-1)*v^u_{k_1}+v^d_{k_1}<(M-1)*v^u_{k_2}+v^d_{k_2}$,

where $d_{v^u_{k_1}v^d_{k_1}}$ represents the initial amount of traffic between the uplink zone $v^u_{k_1}$ and the downlink zone $v^d_{k_1}$.

In Step 1, $z^u_j=1\,(z^d_j=1)$ if the uplink zone j is selected (downlink zone j is selected) and $z^u_j=0\,(z^d_j=0)$ other- wise. The parameter $e_k=1$ if the kth element v_k of V is selected as an element of a switching matrix and $e_k=0$ otherwise. Uplink and downlink zone pairs selected for obtaining the current switching matrix are included in the matrix X. Step (1.1.1) updates the numbers of available transponders and intersatellite links. Step 2 makes a new ordered matrix V using the old ordered matrix V and the matrix X.

The following theorem shows that the computational complexity of the algorithm MDFS is $O(r^2)$ where r is the number of non-zero entries in D, which is smaller than the previous algorithms.

Theorem 2. The worst case overall time complexity of the algorithm MDFS is $O(r^2)$, where r is the number of non-zero entries in D.

Proof. $O(r\log r)$ time is needed to make the ordered matrix V initially. $O(r)$ iterations of steps (1.1) and (2.1) are needed to obtain a switching matrix and to make a new ordered matrix V. In the worst case, the number of switching matrices generated and additional ordering processes are $O(r)$, since at least one non-zero entry is entirely scheduled in each switching matrix. Thus the worst case overall time complexity of MDFS is $O(r\log r+r^2+r^2)=O(r^2)$. ■

4. Simulation Results

Test examples have $S=2,3,4$, $M=6,8,12$. For each case we have applied our algorithms to 100 traffic matrices containing integers randomly generated from a uniform distribution between 0 and u, $u=5,10,20,50$. This random generation format is exactly the same as those in [4] and [10].

<Table 1> shows simulation results when each satellite has M/S transponders and each ordered

Table 1. Computational Results for Regular Systems

S	M	u	LB	HI	SCS	$MDFS_1$	$MDFS_{0.5}$	$MDFS_0$
2	6	5	25.87[a]	26.83[b](3.69[c])	26.21(1.36)	25.95(0.31)	25.89(0.08)	25.89(0.08)
				18.95[d]	19.16	19.34	19.42	19.40
2	6	10	49.79	51.53(3.08)	50.13(0.61)	50.06(0.54)	49.84(0.10)	49.91(0.24)
				25.08	25.00	25.50	25.33	25.44
2	6	20	102.04	105.35(2.66)	102.83(0.56)	102.63(0.58)	102.07(0.03)	102.08(0.04)
				29.81	30.09	30.38	30.17	30.20
2	6	50	245.09	256.11(5.20)	247.26(0.93)	246.94(0.75)	245.18(0.04)	245.19(0.04)
				33.18	33.35	33.61	33.46	33.53
2	8	5	44.62	44.91(0.75)	44.62(0.00)	44.62(0.00)	44.62(0.00)	44.62(0.00)
				31.07	31.66	34.73	34.82	34.82
2	8	10	85.91	86.66(0.66)	85.91(0.00)	85.91(0.00)	85.91(0.00)	85.91(0.00)
				42.00	42.14	45.60	45.81	45.81
2	8	20	176.66	177.44(0.39)	176.66(0.00)	176.96(0.17)	176.66(0.00)	176.66(0.00)
				51.22	51.55	54.30	54.44	54.44
2	8	50	434.86	438.54(0.73)	434.86(0.00)	435.01(0.03)	434.86(0.00)	434.86(0.00)
				57.93	58.22	59.30	59.46	59.46
2	12	5	95.60	96.02(0.35)	95.60(0.00)	95.60(0.00)	95.60(0.00)	95.60(0.00)
				63.31	63.80	66.29	66.21	66.21
2	12	10	189.49	190.27(0.45)	189.49(0.00)	189.49(0.00)	189.49(0.00)	189.49(0.00)
				87.58	87.24	89.83	89.83	89.83
2	12	20	380.04	381.60(0.49)	380.04(0.00)	380.04(0.00)	380.04(0.00)	380.04(0.00)
				109.15	108.01	110.81	110.90	110.90
3	12	5	49.26	52.68(6.58)	51.45(4.94)	49.29(0.06)	49.27(0.22)	49.27(0.02)
				45.97	46.74	42.73	42.96	42.97
3	12	10	95.45	103.44(8.98)	98.95(3.82)	95.78(0.35)	95.45(0.00)	95.45(0.00)
				73.19	73.02	67.91	67.21	67.20
3	12	20	190.70	207.54(8.71)	198.87(4.49)	191.46(0.40)	190.72(0.01)	190.72(0.01)
				99.82	99.30	93.61	93.28	93.39
4	12	5	41.40	47.96(15.34)	42.08(1.69)	42.53(0.31)	41.49(0.22)	41.58(0.43)
				42.50	39.26	34.44	36.73	36.87
4	12	10	79.99	94.72(18.08)	81.17(1.43)	80.17(0.23)	80.18(0.24)	80.25(0.33)
				68.99	65.44	53.36	58.11	58.32
4	12	20	160.76	189.72(17.99)	162.54(1.06)	160.84(0.05)	160.88(0.07)	161.41(0.40)
				95.88	93.20	78.27	85.27	84.30

[a] : average lower bound

[b] : average duration

[c] : average surplus percentage from lower bound(%)

[d] : average number of switching configurations

pair of satellites one ISL, and <Table 2> shows results for more general systems with an arbitrary number intersatellite links and transponders. In the tables, HI and SCS represent the Heuristic ISL algo-

Table 2. Computational Results for General Systems

S	M	u	LB	SCS	$MDFS_1$	$MDFS_{0.5}$	$MDFS_0$
2	6	5	26.49[a]	28.69[b](9.98[c])	26.74(0.94)	26.72(0.87)	26.75(0.98)
				20.21[d]	18.74	18.74	18.89
2	6	10	51.19	56.06(9.83)	51.72(1.04)	51.55(0.70)	51.50(0.61)
				25.93	24.26	24.05	24.40
2	6	20	103.80	113.18(9.38)	105.86(1.98)	104.05(0.24)	104.11(0.30)
				30.62	29.76	29.06	29.39
3	12	5	63.97	66.68(4.14)	64.22(0.39)	64.18(0.33)	64.45(0.75)
				55.82	51.33	51.11	46.99
3	12	10	125.19	130.82(3.76)	125.45(0.21)	125.49(0.24)	125.87(0.54)
				83.84	72.37	72.72	67.66
3	12	20	252.20	261.25(3.38)	252.65(0.18)	253.09(0.35)	253.57(0.54)
				107.39	90.83	92.35	91.15
4	12	5	48.48	51.67(6.61)	48.72(0.50)	48.71(0.47)	48.75(0.56)
				45.13	40.40	42.38	42.78
4	12	10	95.12	101.31(6.20)	95.47(0.37)	95.34(0.23)	95.63(0.54)
				71.17	62.22	65.70	65.49
4	12	20	190.18	203.66(7.73)	190.63(0.24)	190.48(0.16)	191.27(0.57)
				97.42	84.75	88.54	95.90

[a] : average lower bound

[b] : average duration

[c] : average surplus percentage from lower bound(%)

[d] : average number of switching configurations

rithm presented by Ganz *et al.* [4], and the satellite cluster scheduling algorithm presented by Kim *et al.* [10], respectively.

<Table 1> shows that the algorithm MDFS needs much less durations than the algorithms HI and SCS, and $MDFS_{0.5}$ (MDFS with $w=0.5$) minimizes the average durations in all cases except last two cases. The average number of switching configurations could be another important factor in the SS/TDMA scheduling. In cases of three-satellite and four-satellite systems, the proposed algorithm needs much smaller number of switching configurations than HI and SCS.

In <Table 2>, the matrices L's for two, three and four satellites are given as, respectively,

$$\begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}, \quad \begin{pmatrix} 3 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 4 \end{pmatrix} \text{ and } \begin{pmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 1 & 1 \\ 2 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}.$$

<Table 2> also shows that the algorithm MDFS needs much less durations than the algorithm SCS, and $MDFS_{0.5}$ minimizes the average durations in six cases. In addition, in all cases, the proposed algorithm MDFS needs much smaller number of switching configurations than the algorithm SCS.

5. Conclusions

In this paper, a new efficient algorithm to solve the time slot scheduling problem for a satellite cluster was suggested, which is one of the most interesting problems in satellite communication scheduling area. The proposed algorithm MDFS has lower computational complexity than other existing algorithms and provides a solution very close to the optimal schedule.

This type of scheduling will be more important and interesting problems in the future when the low earth orbit satellite communication systems become more commonly used. Thus applications of the proposed algorithm would be an interesting future research work.

References

1. Bertossi, A. A., Bongiovanni, G. and Bonuccelli, M. A., "Time slot assignment in SS/TDMA systems with intersatellite links," *IEEE Transactions on Communications,* Vol. 35, pp. 602-608, 1987.
2. Bongiovanni, G., Coppersmith, D. and Wong, C. K., "An optimum time slot assignment algorithm for an SS/TDMA system with variable number of transponders," *IEEE Transactions on Communications,* Vol. 29, pp. 721-726, 1981.
3. Ganz, A. and Gao, Y., "Efficient algorithms for an SS/TDMA scheduling," *IEEE Transactions on Communications,* Vol. 40, pp. 1367-1374, 1992.
4. Ganz, A. and Gao, Y., "SS/TDMA scheduling for satellite clusters," *IEEE Transactions on Communications*, Vol. 40, pp. 597-603, 1992.
5. Gopal, I. S., Bongiovanni, G., Bonuccelli, M. A., Tang, D. T. and Wong, C. K., "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Transactions on Communications*, Vol. 30, pp. 2475-2481, 1982.
6. Gopal, I. S. and Wong, C. K., "Minimizing the number of switchings in an SS/TDMA system," *IEEE Transactions on Communications*, Vol. 33, pp. 497-501, 1985.
7. Inukai, T., "An efficient SS/TDMA time slot assignment algorithm," *IEEE Transactions on Communications*, Vol. 27, pp. 1449-1455, 1979.
8. Inukai, T., "Comments on analysis of a switch matrix for an SS/TDMA system," *Proceedings of IEEE*, Vol. 66, pp. 1669-1670, 1978.
9. Ito, Y., Urano, Y., Muratani, T. and Yamaguchi, M., "Analysis of a switch matrix for an SS/TDMA system," *Proceedings of IEEE*, Vol. 65, pp. 411-419, 1977.
10. Kim S. and Kim, S. H., "An efficient algorithm for generalized SS/TDMA scheduling with satellite cluster and intersatellite links," *International Journal of Satellite Communications*, Vol. 13, pp. 31-37, 1995.
11. Pomalaza-Raez, C. A., "A note on efficient SS/TDMA assignment algorithms," *IEEE Transactions on Communications*, Vol. 36, pp. 1078-1082, 1988.

장근녕

1988년 서울대학교 국제경제학과 졸업
1990년 한국과학기술원 경영과학과 석사
1994년 한국과학기술원 경영과학과 박사
1994-1996년 한국전자통신연구원 선임 연구원으로 근무
현재 : 신라대학교(구, 부산여자대학교) 경영정보학과 재직
관심분야 : 정보통신시스템 설계, 전자상거래, 정보보안 등